

REMARKS

Claims 1, 5-8, 14, 16-17, 20 and 24-27 have been amended. Claims 1-31 remain pending in the application. Reconsideration is respectfully requested in light of the following remarks.

New Rejections:

The Examiner rejected claims 12 and 30 under 35 U.S.C. § 103(a) as being unpatentable over Webb in view of common art and Hughes in view of common art. Applicants respectfully traverse these rejections for at least the following reasons.

Regarding claim 12, contrary to the Examiner's assertion, the cited references, in view of common art, fail to teach or suggest *the STLF checker is configured to replay one or more additional operations that are dependent on the load operation if the STLF checker detects incorrect operation of the STLF buffer*. The Examiner submits that Hughes and Webb disclose the microprocessor of claim 9 and the method of claim 28. The Examiner further submits that he believes it is inherent that a program must reissue at least some element of the previous instruction to receive a desired output, even though this is not expressed explicitly in either reference (emphasis added). The Examiner asserts that it is common in the art to reissue instructions when the output is not a desired result and that Hughes and Webb would have been motivated to utilize this technique because reissuing instructions is a common, simple, fast, and effective technique for gaining the correct output. The Examiner states that he is not aware of any other technique. The Examiner submits, therefore, that it would have been obvious at the time of the invention for one of ordinary skill in the art to take the processing system of either Hughes or Webb and combine it with the ability to reissue instructions when the original output is not a desired result.

Applicants traverse the Examiner's statements and the Examiner's reliance on allegedly common art. First, Applicants note that claim 12 does not recite reissuing the

previous instruction (or at least some element of the previous instruction) in response to the output being an incorrect result, as the Examiner suggests. Instead, it recites that the STLF checker is configured to replay one or more additional operations that are dependent on a load operation for which the STLF buffer operated incorrectly. Applicants assert that merely reissuing the previous instruction when the output is not a desired result does not teach or suggest the limitation referenced above. Applicants again submit that there is nothing **inherent** about replaying additional, dependent instructions when the previous instruction is reissued, and that there are many ways a system could operate correctly without replaying the dependent instructions. For example, in some systems, any dependent instructions may be prevented from executing until correct operation of an STLF buffer, or a similar mechanism, has been verified for the load operation on which they depend. Therefore, there would be no need to replay the dependent instructions following incorrect operation of the buffer with respect to the load operation. Second, Applicants disagree with the Examiner's characterization that reissuing instructions is a "common, simple, fast, and effective technique for gaining the correct output" and note that the Examiner has not provided any evidence of record to teach or suggest that this is the case.

For at least the reasons above, Applicants assert that the rejection of claims 12 and 30 are not supported by the cited art, and removal thereof is respectfully requested.

Maintained Rejections:

First ground of rejection:

The Examiner rejected claims 1-3, 5, 6, 8-22, 24, 25 and 27-31 under 35 U.S.C § 102(b) as being anticipated by Hughes (WO 01/35212). Applicants respectfully traverse this rejection for at least the following reasons.

In the Response to Arguments section of the Office Action mailed November 28, 2006, the Examiner disagrees with Applicants' arguments that merely using a value to

select an entry cannot be considered “indexing”, and provides a definition of the noun “index”. Applicants note that the Examiner has referred to the noun “index”, not the verb “indexing”. Applicants maintain their traversal of the rejection. However, claim 1 (and independent claims 14 and 20) have been further amended to clarify both the generated index value and the indexing operation of Applicants’ claimed invention. As discussed below, the cited art fails to disclose the limitations of Applicants’ claims involving this generated index value and indexing operation.

Also in the Response to Arguments section of the Office Action mailed November 28, 2006, the Examiner disagrees with Applicants’ arguments regarding “generating an index” and submits that, as interpreted in this Office Action, an index can be generated in at least two ways: 1) When the bits used for the index are determined based on the output of any logic components or 2) taking the pre-existing bits and organizing them in a fashion that they can be utilized as an index. Applicants assert, however, that the Examiner has not cited anything in the art of record supporting these interpretations. Applicants also assert that, as discussed below, the cited art does not teach the limitations of Applicants’ claims, even if this interpretation of “generating an index” is applied to the cited art.

Regarding claim 1, contrary to the Examiner’s assertion, Hughes fails to teach or suggest an indexed STLF (Store-to-Load Forwarding) buffer including *a plurality of entries each of which is selectable using a fixed index value unique to that entry* and a load store unit configured to *generate an index value dependent on at least a portion of an address of a load operation, wherein the generated index value is one of the fixed index values; index into the indexed STLF buffer using the generated index value to select one of the plurality of entries; and forward data included in the one of the plurality of entries selected by the generated index value as a result of the load operation.*

The generated index value of Applicants’ invention is not merely “used to select” an entry, but that it is used to “index into the indexed STLF buffer.” Claim 1, as amended, clarifies this limitation by reciting that the indexed STLP buffer includes a

plurality of entries each of which is selectable using a fixed index value unique to that entry, and that the generated index value is one of the fixed index values. Applicants assert that as amended, the independent claims clearly distinguish over the cited art, since the cited reference uses an index portion of an address in a comparison operation to select an entry in a buffer that uses a variable value, rather than a fixed value, to select each entry in the buffer. The Examiner's citations describe that a comparison is made between an index portion of a load address and an index portion of an entry in an LS2 buffer (or store queue) to determine whether to forward data from the store queue. Applicants assert that there is nothing in Hughes that teaches or suggests that each entry in the buffer is selectable using a fixed index value unique to the entry, nor that a generated index value matching this fixed index value is used to select an entry in the buffer. **Instead, the entries in the LS2 buffer of Hughes are selectable based on the contents of the entry, (i.e., an index portion of the store address contained in the entry) which is clearly a variable value.**

Therefore, for at least the reasons above, the rejection of claim 1 is not supported by the cited art and Applicants respectfully request the withdrawal thereof.

Independent claims 14 and 20 each include limitations similar to those discussed above regarding claim 1. Therefore, the arguments presented above regarding claim 1 apply with equal force to these claims as well.

Regarding claim 5, contrary to the Examiner's assertion, Hughes fails to teach or suggest *the load store unit is configured to select which one of the plurality of entries to allocate to a store operation by generating an additional index value dependent on at least a portion of an address of the store operation*. The Examiner cites FIG. 1 (ADDR – Tag) as teaching this limitation. In the Request for Continuing Examination, Applicants asserted that the address tag portion of an entry in store queue 400 is not described as an additional index value used to select which one of the plurality of entries (i.e., to select which one of the plurality of entries to **allocate to a store operation**.) In his Response, the Examiner cited Hughes, page 7, line 10-14 as disclosing that the tag is used in “selecting

an entry from the STLF buffer.” However, this passage describes the load address, when it may be available for comparison, when the load’s hit signal may be determined, and when the way indication for the load may be determined. **It has absolutely nothing to do with selecting an entry to allocate to a store operation**, (i.e., to select an entry in which to store information about a store operation). Applicants assert that nothing in Hughes discloses the additional index value of claim 5, i.e., an index value generated dependent on at least a portion of an address of the store operation and used to select which one of the plurality of STLF entries (or a store queue entry) is allocated to a store operation.

For at least the reasons above, the rejection of claim 5 is unsupported by the cited art and removal thereof is respectfully requested. Claim 16 and 24 include limitations similar to claim 5, and so the arguments presented above apply with equal force to these claims, as well.

Regarding claim 6, contrary to the Examiner’s assertion, Hughes fails to teach or suggest *the load store unit is configured to generate the additional index value dependent on both the at least a portion of the address of the store operation and a number of bytes of data operated on by the store operation, and wherein the load store unit is configured to generate the index value dependent on both the at least a portion of the address of the load operation and a number of bytes of data operated on by the load operation*. As discussed above regarding claim 5, Hughes fails to teach or suggest the additional index value of claim 5, at all. In the Final Action, the Examiner cited page 8, lines 10-13, as teaching these limitations, “The Examiner asserts that the data size is used as an index to determine if a TRAP signal is to be generated due to a load requesting a larger block of data than was previously stored at that memory location.” However, this citation states, “Hit control circuit 402 may use the offset and size information to determine whether or not to cause the forwarding of data stored in store queue 400 for the load (in addition to the index comparisons, hit bits, and way indications described above).” The offset and size information are used in additional comparison operations to determine if there is a match between a load operation and a previous store operation, not to generate an index

value for selecting an entry in a STLF buffer to allocate to a store operation. This citation has nothing to do with generating such an index value, much less with the specific details for generating the two different index values that are recited in claim 6. Furthermore, the Examiner's remarks regarding a TRAP signal have absolutely no basis in the cited art, as Hughes does not include a description of a TRAP signal at all. These remarks appear to be copied from the Examiner's previous rejection of claim 6 as being anticipated by Webb and are therefore improper to include in the rejection of claim 6 as being anticipated by Hughes. In addition, the use of size data to determine if a TRAP signal is generated has nothing to do with the limitations of claim 6, which are directed toward generating index values used for selecting an entry in a STLF buffer. Therefore Hughes clearly does not anticipate claim 6.

Applicants also note that the Examiner failed to address Applicants' arguments above in the Office Action mailed November 28, 2006.

For at least the reasons above, the rejection of claim 6 is unsupported by the cited art and removal thereof is respectfully requested. Claim 17 and 25 include limitations similar to claim 6, and so the arguments presented above apply with equal force to these claims, as well.

Regarding claim 8, contrary to the Examiner's assertion, Hughes fails to teach or suggest *the additional index value comprises a portion of the address targeted by the store operation*. The Examiner again cited FIG. 1 (ADDR – tag) as teaching this limitation. However, as discussed above, Hughes does not teach generating the additional index value of claim 5 at all, much less one comprising a portion of the address targeted by the store operation. The address tag in FIG. 1 is the portion of the store address that is stored as a tag by the data cache, not an additional index value used in selecting an entry in a STLF buffer to allocate to a store operation, as required by Applicants' claims 5 and 8.

Applicants also note that the Examiner failed to address Applicants' arguments above in the Office Action mailed November 28, 2006.

For at least the reasons above, the rejection of claim 8 is unsupported by the cited art and removal thereof is respectfully requested. Claim 27 includes limitations similar to claim 8, and so the arguments presented above apply with equal force to this claim, as well.

Regarding claim 10, contrary to the Examiner's assertion, Hughes fails to teach or suggest *the STLF checker is configured to perform an associative address comparison to identify all issued store operations targeting a same address as the load operation and to implement a find-first algorithm to select a youngest issued store operation that is older than the load operation*. In the Final Action, the Examiner cited page 8, lines 14-17 as teaching these limitations. However, while this passage states, "Hit control circuit 402 may determine the youngest (most recently executed) store in program order among the stores corresponding to entries which are hit and may forward the data from that entry," there is nothing in Hughes that discloses the specific limitations of claim 10. For example, there is no mention of implementing a find-first algorithm, as recited in claim 10. Therefore Hughes cannot be said to anticipate claim 10.

In the Response to Arguments section of the Office Action mailed November 28, 2006, the Examiner disagrees with Applicants' arguments above, stating, "It is unclear what definition Applicant believes the claimed invention requires from the term "find-first algorithm", but it appears sufficient to say that the remainder of the functionality of claim 10 can be considered a "find-first algorithm" and Applicant says nothing to suspect the contrary." Applicants again submit that while the Examiner's passage describes a similar goal as that recited in claim 10, "to select a youngest issued store operation that is older than the load operation", it does not describe how this determination is implemented. Claim 10 recites that this is implemented using a find-first algorithm. Hughes, however is silent as to the specifics of the determination. Applicants assert that many different methods may be implemented to select the youngest of the operations,

depending on the order in which the entries are stored (e.g., whether they are stored in program order or execution order), what additional information may be stored within each entry (e.g., an indicator of age, or program order), or other implementation details. Therefore, the “youngest” of the operations may not be the “first” of the operations that correspond to a hit, and may not be selectable using a “find-first algorithm.” The Examiner’s other cited reference, for example, uses an INUM field to indicate the relative age of various entries in a store queue.

For at least the reasons above, the rejection of claim 10 is unsupported by the cited art and removal thereof is respectfully requested. Claim 28 includes limitations similar to claim 10, and so the arguments presented above apply with equal force to this claim, as well.

Regarding claim 11, contrary to the Examiner’s assertions, Hughes fails to teach or suggest *the STLF checker is configured to replay the load operation if the STLF checker identifies incorrect operation of the STLF buffer*. The Examiner cites page 5, lines 15-18 as teaching this limitation. However, as discussed in Applicants’ Request for Continuing Examination, this passage describes what happens in response to a cache miss for the load operation, not in response to an STLF checker identifying incorrect operation of the STLF buffer. Therefore, Hughes clearly does not anticipate claim 11.

In the Response to Arguments section of the Office Action mailed November 28, 2006, the Examiner notes that the passage above refers to the fact that “the load may be reattempted”, and submits that the fact that this passage describes that a reattempted load may result in a cache hit does not prevent the remaining times that this is not the case from anticipating the claim. Applicants assert, however, that none of the Examiner’s citations in Hughes describe replaying a load operation in response to identifying incorrect operation of the STLF buffer.

For at least the reasons above, the rejection of claim 11 is unsupported by the cited art and removal thereof is respectfully requested. Claim 19 and 29 include

limitations similar to claim 11, and so the arguments presented above apply with equal force to these claims, as well.

Regarding claim 12, contrary to the Examiner's assertion, Hughes fails to teach or suggest *the STLF checker is configured to replay one or more additional operations that are dependent on the load operation if the STLF checker detects incorrect operation of the STLF buffer*. The Examiner again cited page 5, lines 15-18 as teaching this limitation. However, as discussed above, this passage does not describe replaying a load operation in response to the STLF checker detecting incorrect operation of the STLF buffer, much less replaying one or more additional operations, as the Examiner suggests. Instead, it describes actions that may occur following a miss in the data cache for a load instruction. The Examiner asserted, "...after an initial load instruction is attempted and fails, any other instructions which have been issued which depend on the load instruction for data must be replayed once the data has been made available through replay of the load operation." Applicants submitted that this is purely speculation by the Examiner and that there are many different ways to recover from a data cache miss for a load instruction, not all of which necessarily involve replaying the load instruction or any dependent instructions. Furthermore, a data cache miss for a load operation is not necessarily caused by incorrect operation of a STLF buffer. The Examiner has not cited anything in Hughes to teach or suggest that a load operation and one or more additional operations are replayed in response to incorrect operation of a STLF buffer (or Hughes' store queue 400 or LS2 62), nor is there anything in Hughes describing such replays. Therefore, Hughes clearly does not anticipate claim 12.

In the Response to Arguments section of the Office Action mailed November 28, 2006, the Examiner refers to his remarks regarding Webb and states that an Official Notice rejection has been added in attempt to address Applicants' concern. Applicants note, however, that the Examiner did not state an Official Notice rejection of claim 12. Instead, claim 12 was rejected in light of the cited references "in view of common art" without the taking of Official Notice. Moreover, Applicants traverse the Examiner's

“common art” as discussed above. The Examiner’s additional remarks regarding Webb are addressed below.

For at least the reasons above, the rejection of claim 12 is unsupported by the cited art and removal thereof is respectfully requested. Claim 30 includes limitations similar to claim 12, and so the arguments presented above apply with equal force to this claim, as well.

Second ground of rejection:

The Examiner rejected claims 4 and 23 under 35 U.S.C. § 103(a) as being unpatentable over Hughes, and claims 7 and 26 as being unpatentable over Hughes in view of Hennessy (“Computer Organization and Design”). Applicants respectfully traverse these rejections for at least the following reasons.

Regarding claim 4, contrary to the Examiner’s assertion, Hughes fails to teach or suggest *each of the plurality of entries in the STLF buffer has a capacity to store a maximum amount of data that can be written by a store operation*. In the Final Action, the Examiner admits that Hughes fails to disclose this limitation, and notes “Hughes does not disclose the data bus width of the processor of his invention.” Applicant again asserts that the bus width is irrelevant, and notes that claim 4 recites nothing about a bus width.

In the Final Action, the Examiner took Official Notice that data buses of size 8, 16, 32, or 64 bits are extremely well known in the art and that with any of these data buses in place in Hughes invention, the data buffer would be able to hold at least the maximum amount of data specified by a data store operation. Applicants asserted however, that the maximum amount of data that can be written by a store operation is not necessarily the same as, or dependent on, the data bus width, but instead may be dependent on the instruction set architecture (i.e., the instruction sets of different processors may provide for store operations involving different numbers of bytes of data and/or the implementation of the instruction set may provide for different numbers of bus

cycles for each store operation). Therefore, Applicants asserted that the Examiner's Official Notice does not teach or suggest the limitations of claim 4.

In the Response to Arguments section of the Office Action mailed November 28, 2006, the Examiner refers to remarks made regarding Webb and regarding claim 7 (below.) In these remarks, the Examiner asserts that the claim language calls for "a capacity" which is defined as "the ability to receive or contain" by the American Heritage Dictionary, and that the appropriately sized data bus in an ability to receive. Applicants assert, however, that the Examiner is ignoring the plain language of the claim, which explicitly recites, "a capacity to store..." not just "a capacity". Therefore, the claim is clearly not directed to a capacity to receive, as suggested by the Examiner. Applicants again assert that, as discussed above, *a capacity to store a maximum amount of data that can be written by a store operation* is not defined merely by a data bus size, that the Examiner's remarks regarding a data bus width are irrelevant, and that the limitation discussed above is not taught by Hughes or Webb.

For at least the reasons above, the rejection of claim 4 is not supported by the cited art and removal thereof is respectfully requested. Claim 23 includes limitations similar to claim 4, and so the arguments presented above apply with equal force to this claim, as well.

Regarding claim 7, contrary to the Examiner's assertion, Hughes in view of Hennessey fails to teach or suggest *the additional index is generated by right-shifting a lower portion of the address targeted by the store operation by an amount equal to a logarithm in base two of the number of bytes of data operated on by the store operation*. The Examiner admits that Hughes fails to disclose this limitation and relies on Hennessey to teach it, "Hennessey discloses indexing a cache by means of the lower portion of an address, minus the appropriate offset for minimum memory access size (byte, in Hennessey's case)... Right shifting is an extremely well-known method of eliminating undesired bits to the right of desired bits. Hennessey teaches that removing the two bits for the byte offset reduces the total cache size, as fewer bits must be kept in the tag entry

field of each entry. A smaller cache size takes up less space on chip and is less expensive to implement.” The Examiner submits that it would have been obvious to one of ordinary skill in the art at the time of invention to have included the method of generating a cache index disclosed by Hennessey in Hughes’ invention for the benefit of reducing the necessary cache size. Applicants assert, however, that claim 7 has nothing to do with indexing a cache or reducing a cache size, but is directed toward a specific method for generating an additional index value for selecting an entry in the STLF buffer to allocate to a store operation. **As discussed above, Hughes does not teach this additional index value, so one would not have any reason to explore ways to generate such an index value in the system of Hughes.** Furthermore, if the method of Hennessey were applied to Hughes’ invention, the result would not teach the limitations of claim 7. Instead, they would result in a change in the way the data cache, not the store queue, was organized and indexed.

In the Response to Arguments section of the Office Action mailed November 28, 2006, the Examiner disagrees with the above argument, citing his remarks regarding Webb. In these remarks, the Examiner asserts that Webb teaches the additional index value. Therefore, these remarks have nothing to do with Hughes and are improperly applied to this rejection.

For at least the reasons above, the rejection of claim 7 is not supported by the cited art and removal thereof is respectfully requested. Claim 26 includes limitations similar to claim 7, and so the arguments presented above apply with equal force to this claim, as well.

Previously Maintained Rejections:

The Examiner rejected claims 1-3, 5, 6, 8-12, 14-22, 24, 25 and 27-30 under 35 U.S.C § 102(b) as being anticipated by Webb, et al. (U.S. Patent 6,360,314) (hereinafter “Webb”). Applicants respectfully traverse this rejection for at least the following reasons.

Regarding claim 1, contrary to the Examiner's assertion, Webb fails to teach or suggest an indexed STLTF (Store-to-Load Forwarding) buffer including *a plurality of entries each of which is selectable using a fixed index value unique to that entry* and a load store unit configured to *generate an index value dependent on at least a portion of an address of a load operation, wherein the generated index value is one of the fixed index values; index into the indexed STLTF buffer using the generated index value to select one of the plurality of entries; and forward data included in the one of the plurality of entries selected by the generated index value as a result of the load operation.*

First, as noted in the Request for Continuing Examination, Webb does not disclose a load store unit configured to store information associated with load and store operations, nor including an indexed Store-to-Load Forwarding (STLTF) buffer having the limitations of claim 1. The Examiner refers to FIG. 4, including buffer 428 and queue 426 as teaching these limitations. However, FIG. 4 clearly illustrates that store data buffer 428 and store queue 426 are components of data cache subsystem 420, and not load/store unit 418. In addition, there is nothing in Webb describing that data included in the entry selected by the generated index value is forwarded as a result of the load operation, as required by claim 1. Instead, data from store data buffer 428, which is not selected using the generated index value to index into the store data buffer, is forwarded as a result of a load operation in which the tag in the store queue and the tag reference match for a particular entry in the store queue. In Webb, it is a store queue entry that may be selected by indexing into the queue using a generated index value, not a store data buffer entry.

In the Response to Arguments section of the Office Action mailed November 28, 2006, the Examiner submits that the STLTF buffer and associated logic constitute part of the load/store unit as they all assist in performing load and store operations. Applicants assert, however, that even if store data buffer 428 and store queue 426 were considered part of the load/store unit, they do not teach or suggest all the limitations of the STLTF buffer recited in claim 1, specifically that each entry of the STLTF buffer is selectable

using a fixed index value unique to that entry, using a generated index value to index into the STLTF buffer to select an entry, and forwarding data included in the entry as a result of the load. Instead, Webb discloses that data is forwarded from the store data buffer if the tag in the store queue and the tag reference match for a particular entry.

Further regarding claim 1, Webb does not disclose indexing into the indexed STLTF buffer using the generated index value to select one of the entries in the STLTF buffer and forwarding data from the selected entry. The Examiner previously cited column 5, lines 5-8 and 19-22, as teaching these limitations. However, there is nothing in these citations that teaches that the load store unit, or any other apparatus, is configured to **generate** an index value. They only describe the use of an index. Furthermore, the index referred to in these citations is not a generated index value used to index into an indexed STLTF buffer, or to Webb's store data buffer 428, which the Examiner equates with Applicants' STLTF buffer. Instead, these citations describe indexing into dcache unit 430, and one of its component (tag store 432.)

In the Response to Arguments section of the Office Action mailed November 28, 2006, the Examiner submits that Webb also describes indexing into the STL buffer as well, in column 6, lines 6-10. Applicants assert that this passage describes indexing into store queue 426, but does not disclose forwarding data from the entry in 426 (i.e., the entry selected by indexing into store queue 426). Instead, as described above, it describes forwarding data from store data buffer 428 if the tag in store queue 426 and the tag reference match for the selected entry in store queue 426.

As the Examiner is no doubt aware, anticipation requires the presence in a single prior art reference disclosure of each and every limitation of the claimed invention, arranged as in the claim. M.P.E.P 2131; *Lindemann Maschinenfabrik GmbH v. American Hoist & Derrick Co.*, 221 USPQ 481, 485 (Fed. Cir. 1984). The **identical** invention must be shown in as complete detail as is contained in the claims. *Richardson v. Suzuki Motor Co.*, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). As discussed above, Webb fails to disclose forwarding data from an entry selected by indexing into a STLTF buffer using a generated

index value, as required by claim 1. Therefore, Webb cannot be said to anticipate claim 1.

For at least the reasons above, the rejection of claim 1 is not supported by the cited art and removal thereof is respectfully requested. Independent claims 14 and 20 include limitations similar to those discussed above regarding claim 1, and so the arguments presented above apply with equal force to this claim, as well.

Regarding claim 5, contrary to the Examiner's assertion, Webb fails to teach or suggest *the load store unit is configured to select which one of the plurality of entries to allocate to a store operation by generating an additional index value dependent on at least a portion of an address of the store operation*. The Examiner previously cited column 5, lines 5-8 and 19-23 as teaching this limitation. However, these passages does not describe generating an additional index value that is used to select an entry to allocate to a store operation (or any two indices, for that matter). These two passages refer to the same virtual index, used to index into dcache unit 430 (specifically tag store 432) and store queue 426, as discussed above.

In the Response to Arguments section of the Office Action mailed November 28, 2006, the Examiner submits that the additional index value as claimed "is required to depend on a portion of the address of the store operation (col. 5 lines 5-8 and 19-23) and data operated on by the store operation (col. 6 lines 51-59)." Applicants disagree. Claim 5 does not require the additional index value to be dependent on anything other "at least a portion of an address of the store operation." The Examiner goes on to state that the combination of this data is considered to be "an additional index". This statement is completely unsupported in the cited art. The Examiner's citation in column 6 teaches that in Webb, additional fields may be compared to determine a hit (e.g., a size field and/or an INUM field.) Neither of these constitutes an index value at all, much less an additional index value for use in selecting an entry to allocate to a store operation, and neither is dependent on a portion of an address, as required by claim 5. Applicants assert,

therefore, that this passage has nothing to do with the additional index value of Applicants' claim 5.

For at least the reasons above, the rejection of claim 5 is not supported by the cited art and removal thereof is respectfully requested. Claims 16 and 24 include limitations similar to claim 5, and so the arguments presented above apply with equal force to these claims, as well. Applicants note that the Examiner rejected claim 22 along with claim 5 in his remarks. Applicants assume that this was a typographical error, as claim 24 (and not claim 22) recites limitations similar to those in claim 5.

Regarding claim 6, contrary to the Examiner's assertion, Webb fails to teach or suggest *the load store unit is configured to generate the additional index value dependent on both the at least a portion of the address of the store operation and a number of bytes of data operated on by the store operation, and wherein the load store unit is configured to generate the index value dependent on both the at least a portion of the address of the load operation and a number of bytes of data operated on by the load operation.* The Examiner again cited column 5, lines 5-8 and 19-23, as teaching that the load store unit is configured to generate this index value. However, as discussed above regarding claim 5, Webb does not teach this limitation. The Examiner cites column 6, lines 51-59, as teaching that the index value is generated dependent on the number of bytes of data operated on by the store operation. While this citation describes a comparison between the size field 48 of a store queue entry and the size information of an issuing load, it does not describe generating an index value into an STLF buffer to select an entry dependent on this information and indexing into the buffer using the generated value. The Examiner asserts that the "data size is used as an index to determine if a TRAP signal is to be generated..." However, as would be understood by one of ordinary skill in the art, using a comparison of two data size values to generate a signal is clearly not the same as generating an index value dependent on a data size value. Applicants assert that this TRAP signal is completely irrelevant to the limitations of claim 6.

In the Response to Arguments section of the Office Action mailed November 28, 2006, the Examiner disagrees and submits, “The indexes are matched against every entry in the store queue. The fact that a signal is generated in response to this indexing is immaterial.” Applicants assert, however, that Applicants claims require that the indexing recited in Applicants’ claims is not performed by matching indexes against every entry in the store queue, as suggested by the Examiner, and as described by Webb. Instead, entries are selected by indexing into the STLF buffer to select an entry selectable by a fixed, unique index value. Furthermore, as discussed below, the combination of elements referenced by the Examiner does not perform the function recited in Applicants’ claims for the additional index value.

In the Response to Arguments section of the Final Action, the Examiner stated, “As previously cited and applied, the examiner clarifies that the size of the pending load *is* used as an index to verify the proper data is being loaded from the buffer” and that, “the size of the load is inherently passed into the buffer logic from the load/store unit and compared, as described in column 6, lines 51-59.” Applicants note that claim 6 has nothing to do with “verifying the proper data is being loaded from the buffer” but instead is directed toward indexes used to select an entry in a STLF buffer to allocate to a store operation and to select the store operation by indexing into the buffer. The Examiner further states, “both the address portion and the data access size combined constitute the index described in claim 6.” Applicants disagree. Nothing in Webb teaches or suggests these elements of a load operation, taken separately or in combination with each other, is an additional index value generated by the load store unit and used to select an entry in a STLF buffer to allocate to a store operation or to select the store operation by indexing into the buffer. Only one virtual index is described in Webb and it is not used select an entry to allocate to a store operation (i.e., to storing data from a store operation to be forwarded in response to a load) or to select the store operation by indexing into the buffer.

The Examiner also noted, “Webb performs a comparison of the size as an index in the same manner as the applicant describes in paragraph 36 on page 12 of the

specification” (emphasis added). Applicants note that Webb does not describe performing a comparison to generate an index value for selecting an entry in a STLF buffer, as the Examiner suggests, or using the size itself to select an entry in a STLF buffer. Applicants also note that paragraph 36 of Applicants’ specification describes a comparison between the address or size of a selected entry (i.e., one selected according to a first generated index value) and the address or size of a load operation. It does not describe that this comparison constitutes generating an index value or selecting an entry, as the Examiner suggests. Paragraph 37, however, goes on to describe an example in which size data may be used in generating an index value. This passage describes that an index may be “generated by performing some transformation function on the address of an operation. Indexes may be generated such that any given address and data size pair maps to a single index...” There is nothing in Webb that describes generating an index value dependent on the number of bytes of data operated on by a load or store operation, as required by claim 6, in this or any other manner.

For at least the reasons above, the rejection of claim 6 is not supported by the cited art and removal thereof is respectfully requested. Claims 17 and 25 include limitations similar to claim 6, and so the arguments presented above apply with equal force to these claims, as well.

Regarding claim 8, contrary to the Examiner’s assertion, Webb fails to teach or suggest *the additional index value comprises a portion of the address targeted by the store operation*. The Examiner again cites column 5, lines 19-22, as teaching this limitation. However, this citation describes indexing (using a portion of an address) into dcache unit 430, not into a STLF buffer, or into Webb’s store queue 426 and store data buffer 428. Using an address to index into dcache unit 430 teaches nothing about the composition of the additional index value (for selecting an entry in the STLF buffer) referred to in Applicants’ claim 8.

In the Response to Arguments section, the Examiner “points to FIG. 4 indicating that the additional index value generated by the load/store unit is connected to the buffer

store queue 426 in addition to dcache 430, as indicated by lines 442 and 446.” However, FIG. 4 illustrates that the same index as the one the Examiner refers to in his rejection of claim 1 (described in column 5, lines 5-8 and 19-22) is connected to both the dcache and the buffer store queue. Therefore, this cannot be considered “an additional index value” as the Examiner suggests.

Applicants note that the Examiner failed to address Applicants’ arguments above in the Office Action mailed November 28, 2006.

For at least the reasons above, the rejection of claim 8 is not supported by the cited art and removal thereof is respectfully requested. Claim 27 includes limitations similar to claim 8, and so the arguments presented above apply with equal force to this claim, as well.

Regarding claim 9, contrary to the Examiner’s assertion, Webb fails to teach or suggest *the load store unit further comprises a STLF checker configured to verify operation of the STLF buffer*. The Examiner cites column 7, lines 5-8, as teaching this limitation, and asserts that Webb’s invention contains a unit which verifies operation, specified as the apparatus disclosed in column 7, lines 5-8. However, the Examiner’s citation describes only that, “The present invention provides a methodology and apparatus for determining which of the multiple stores should be used in bypassing the dcache unit 430.” This does not describe a unit *configured to verify operation of the STLF buffer*, as recited in claim 9, but describes only one of the operations of “a methodology and apparatus” for choosing which of multiple stores should be used. Applicants assert that choosing one of multiple stores has nothing to do with verifying operation of the STLF buffer, as recited in claim 9.

In the Response to Arguments section of the Office Action mailed November 28, 2006, the Examiner disagrees and states, “This determination is considered to be verifying the operation of the STLF buffer. Examiner believes this is a fair and reasonable interpretation. The fact that the citation only describes “one of the operations”

does not prevent it from anticipating the clause of claim 9. The claim does not specify that all operation is verified.” **Applicants assert, however, that this passage does not describe that any operation is verified.** Instead it describes that the present invention provides a way to determine which store should be used (i.e., it describes that the present invention will choose the correct store, not that it will verify that it choose the correct store.)

In the Response to Arguments section of the Final Action, the Examiner submits, “there must inherently exist logic to verify that the correct entry is forwarded when multiple stores are pending for the same address location. If the youngest store is not forwarded upon a subsequent load, incorrect operation may result.” Applicants note that the data bypass method and apparatus described at the Examiner’s citation in Webb includes logic to “attempt to provide the data of the most recent store to a subsequently issuing load and thereby avoid getting older data to the load,” (column 7, lines 18-20). Attempting to provide the correct data is clearly not equivalent to the limitation of Applicants’ claim 9, which recites *the load store unit further comprises a STLF checker configured to verify operation of the STLF buffer*. Furthermore, Applicants remind the Examiner that “To establish inherency, the extrinsic evidence ‘must make clear that the missing descriptive matter is **necessarily present** in the things described in the reference, and that it would be so recognized by persons of ordinary skill. Inherency, however, may not be established by probabilities or possibilities. The mere fact that a certain thing may result from a given set of circumstances is not sufficient.” *In re Robertson*, 169 F.3d 743, 745, 49 USPQ2d 1949, 1950-51 (Fed. Cir. 1999) (emphasis added). The Examiner has not cited anything in Webb that teaches or suggests *the load store unit further comprises a STLF checker configured to verify operation of the STLF buffer*.

In the Response to Arguments section of the Office Action mailed November 28, 2006, the Examiner asserts that the use of inherency is correct. The Examiner states, “The functionality considered to be inherent is required for proper functionality of the invention as disclosed. If incorrect entries were forwarded, the processor would be broken and unable to function as the reference states.... There must be logic to insure

proper verification.” Applicants disagree and assert that the invention of Webb could be implemented in such a way that incorrect entries cannot be forwarded (e.g., in such a way as to not forward an entry unless its correct selection is deterministic). In such embodiments, there would be no need to verify this operation.

For at least the reasons above, the rejection of claim 9 is not supported by the cited art and removal thereof is respectfully requested. Claim 18 includes limitations similar to claim 9, and so the arguments presented above apply with equal force to this claim, as well.

Regarding claim 11, contrary to the Examiner’s assertion, Webb fails to teach or suggest *the STLF checker is configured to replay the load operation if the STLF checker identifies incorrect operation of the STLF buffer*. The Examiner previously cited column 1, lines 34-38 as teaching this limitation and asserts that if a load operation does not result in data being provided from the cache, the memory load is replayed to the main memory. This is incorrect. First, this citation teaches nothing about replaying a load operation. It merely describes that the data for a load is retrieved from main memory, instead of from the cache, if it is not found in the cache. Furthermore, this citation has nothing to do with a STLF checker identifying incorrect operation of the STLF buffer. In fact, this citation has nothing to do with the STLF buffer at all. Instead, as stated above, this citation describes only that if data is not found in the cache, it is retrieved from main memory.

Applicants note that the Examiner failed to address Applicants’ argument above that this citation has nothing to do with incorrect operation of an STLF buffer in the Office Action mailed November 28, 2006. In fact, this citation specifically describes the operation of a data cache in a prior art system, i.e., in one that does not include the bypass mechanism of Webb’s invention. Therefore, it cannot teach anything about an STLF checker replaying an operation in response to identifying incorrect operation of the STLF buffer, as required by claim 11, as no such buffer exists.

In the Response to Arguments section of the Final Action, the Examiner submitted, “when a load misses after a first attempt in the buffer of Webb’s system, the load must be replayed (attempted again) to another memory device (in this case, the dcache unit 430). There is nothing is applicant’s specification defining the term “replay” and hence, it has been awarded its broadest reasonable definition.” Applicants disagree with the Examiner’s interpretation of the term “replay” and also with his assertion that Applicants’ specification does not include anything defining this term. Applicants noted that the specification includes a specific example of how a replay may be caused by the STLTF checker on page 16, lines 17-18, “The STLTF checker 303 may cancel load operations that either incorrectly forwarded in STLTF buffer 305 or which incorrectly did not forward in STLTF buffer 305. In such situations, the STLTF checker 303 may cause the load operation to be replayed (e.g., by providing a signal to the scheduler 118).” Thus, it is clear from Applicants’ specification that the term “replay” was intended to refer to situations in which a load operation is canceled and then rescheduled for execution (i.e., the current instance of the load operation is cancelled and another instance of the load operation is scheduled for execution) rather than to situations in which a current load operation may search more than one level of memory in an attempt to fulfill the load.

In the Response to Arguments section of the Office Action mailed November 28, 2006, the Examiner submits that the portion of the specification provided is a suitable definition of the term “replay”. The Examiner submits that when the specification says, “e.g., by providing a signal to the scheduler 118”, this is referring to an example of how the word replay may be used, not to a particular definition for the word. Applicants disagree and also note that the Examiner has not cited any evidence in the art of record that his interpretation of the word “replay” is a reasonable one.

In addition, **Applicants again assert that retrieving data for a load from main memory after a cache miss does not inherently require replaying a load operation (or attempting it again) as the Examiner suggests.** Instead, many processors having a cache initiate an access main memory in parallel with an access in the cache, so that if the

data is not found in the cache, retrieval from main memory has already been initiated and is not delayed until a hit or miss indication is received. In these cases, the load operation would clearly not have to be replayed (or attempted again). It would have already been initiated and would just be allowed to complete.

For at least the following reasons, the rejection of claim 11 is not supported by the cited art and removal thereof is respectfully requested. Claims 19 and 29 include limitations similar to claim 11, and so the arguments presented above apply with equal force to these claims, as well.

Regarding claim 12, contrary to the Examiner's assertion, Webb fails to teach or suggest *the STLF checker is configured to replay one or more additional operations that are dependent on the load operation if the STLF checker detects incorrect operation of the STLF buffer*. The Examiner cites column 7, lines 63-65, as teaching this limitation and asserts that after the in-flight instructions are killed they must inherently be reissued. The Examiner contends that if the instructions are not reissued, the program may produce undesired output or simply cease operation. The Examiner's citation describes that a TRAP signal 454 is provided by store queue 426 to indicate that the bypass mechanism did not provide data when it should have and therefore, the in-flight instructions should be killed. Applicants assert that the system of Webb could respond to the TRAP signal in any number of ways to produce the correct output or prevent the program from ceasing operation. There is nothing in Webb that teaches or suggests that the response to this signal is to *replay one or more additional operations that are dependent on the load operation*, as recited in claim 12. As there is no detailed description of the scheduling of instructions in Webb, or of any **replay** mechanism at all, the Examiner's assertion that one or more additional operations that are dependent on the load operation must be replayed is mere speculation, not a necessary condition of operation.

In the Response to Arguments section of the Final Action, the Examiner submits, "inflight instructions which are killed must inherently be re-issued if proper program outcome is to be achieved. As described by Webb, when the TRAP signal is issued in the

instance described in col. 7, it signals that the load operation did not provide data when it should have (column 7, line 34) resulting in incorrect and/or unavailable data. A program contains instructions in a certain order designed to achieve a pre-defined function. If a system were to kill the inflight instructions following a data miss (like that described in col. 7) without reissuing them, the function of the program may not be achieved.”

Applicants again remind the Examiner that, “To establish inherency, the extrinsic evidence ‘must make clear that the missing descriptive matter is **necessarily present** in the things described in the reference, and that it would be so recognized by persons of ordinary skill. Inherency, however, may not be established by probabilities or possibilities. The mere fact that a certain thing may result from a given set of circumstances is not sufficient.’” *In re Robertson*, 169 F.3d 743, 745, 49 USPA2d 1949, 1950-51 (Fed. Cir. 1999) (emphasis added). As discussed above, the system of Webb would not **necessarily** need to replay one or more additional operations that are dependent on the load operation if the STLFF checker detects incorrect operation of the STLFF buffer in order to recover from instructions being killed and/or to achieve the function of the program.

In the Response to Arguments section of the Office Action mailed November 28, 2006, the Examiner disagrees and invites Applicants to suggest other possibilities. Applicants note, however, that the burden is on the Examiner to point out how the extrinsic evidence makes it clear that the missing descriptive matter is necessarily present.

In the Response to Arguments section of the Office Action mailed November 28, 2006, the Examiner further submits that, “in an effort to alleviate Applicant’s concern on this issue, Official Notice has been taken with regard to claim 12.” Applicants note, however, that as discussed above regarding Hughes, the Examiner did not take Official Notice with regard to claim 12.

For at least the reasons above, the rejection of claim 12 is not supported by the cited art and removal thereof is respectfully requested. Claim 30 includes limitations

similar to claim 12, and so the arguments presented above apply with equal force to this claim, as well.

The Examiner rejected claims 4 and 23 under 35 U.S.C. § 103(a) as being unpatentable over Webb, and claims 7 and 26 as being unpatentable over Webb in view of Hennessy (“Computer Organization and Design”). Applicants traverse this rejection for at least the following reasons.

Regarding claim 4, contrary to the Examiner’s assertion, Webb fails to teach or suggest *each of the plurality of entries in the STLF buffer has a capacity to store a maximum amount of data that can be written by a store operation*. The Examiner admits that Webb fails to disclose this limitation and notes that while Webb discloses the data entry to hold any of a quadword, longword, word, or byte (column 4, lines 66-67), it does not disclose the data bus width of the processor of his invention. The Examiner takes Official Notice that data buses of size 8, 16, 32, or 64 bits are extremely well known in the art and that with any of these data buses in place in Webb’s invention, the data buffer would be able to hold at least the maximum amount of data specified by a data store operation. Applicants note that claim 4 does not recite a limit on the data bus width of the invention, but instead recites *each of the plurality of entries in the STLF buffer has a capacity to store a maximum amount of data that can be written by a store operation*. Applicants assert that the maximum amount of data that can be written by a store operation is not necessarily the same as, or dependent on, the data bus width, but instead may be dependent on the instruction set architecture (i.e., the instruction sets of different processors may provide for store operations involving different numbers of bytes of data and/or the implementation of the instruction set may provide for different numbers of bus cycles for each store operation). Therefore, Applicants assert that the Examiner’s Official Notice does not teach or suggest the limitations of claim 4.

The Examiner submits, “A data bus having a given size less than or equal to 64 bits is beneficial in a processor in that costs of implementation are limited. Larger data

buses require processor components to also grow in size, increasing processor area, power consumption and cost. Implementing a data bus less than or equal to 64 bits in Webb's invention would have been obvious at the time of invention to one of ordinary skill in the art for the benefit of limiting costs, size and power consumption." As noted above, however, claim 4 does not recite "a data bus less than or equal to 64 bits" or anything about a data bus width at all. Even if the data bus of Webb were less than or equal to 64 bits, this would teach nothing about the capacity of the entries of an STLF buffer, as recited in claim 4.

In the Response to Arguments section of the Office Action mailed November 28, 2006, the Examiner asserts that the claim language calls for "a capacity" which is defined as "the ability to receive or contain" by the American Heritage Dictionary, and that the appropriately sized data bus in an ability to receive. Applicants assert, however, that the Examiner is ignoring the plain language of the claim, which explicitly recites, "a capacity to store..." not just "a capacity". Therefore, the claim is clearly not directed to a capacity to receive, as suggested by the Examiner. Applicants again assert that, as discussed above, *a capacity to store a maximum amount of data that can be written by a store operation* is not defined merely by a data bus size, that the Examiner's remarks regarding a data bus width are irrelevant, and that the limitation discussed above is not taught by Webb.

For at least the reasons above, the rejection of claim 4 is not supported by the cited art and removal thereof is respectfully requested. Claim 23 includes limitations similar to claim 4, and so the arguments presented above apply with equal force to this claim, as well.

Regarding claim 7, contrary to the Examiner's assertion, Webb in view of Hennessy fails to teach or suggest *the additional index value is generated by right-shifting a lower portion of the address targeted by the store operation by an amount equal to a logarithm in base two of the number of bytes of data operated on by the store operation*. The Examiner admits that Webb fails to disclose this limitation and relies on

Hennessey to teach it, “Hennessey discloses indexing a cache by means of the lower portion of an address, minus the appropriate offset for minimum memory access size (byte, in Hennessey’s case)... Right shifting is an extremely well-known method of eliminating undesired bits to the right of desired bits. Hennessey teaches that removing the two bits for the byte offset reduces the total cache size, as fewer bits must be kept in the tag entry field of each entry. A smaller cache size takes up less space on chip and is less expensive to implement.” The Examiner submits that it would have been obvious to one of ordinary skill in the art at the time of invention to have included the method of generating a cache index disclosed by Hennessey in Webb’s invention for the benefit of reducing the necessary cache size. Applicants assert, however, that claim 7 has nothing to do with indexing a cache or reducing a cache size, but is directed toward a specific method for generating an additional index value for selecting an entry in the STLF buffer. As discussed above, Webb does not teach generating this additional index value, so one would not have any reason to explore ways to generate such an index value. Furthermore, if the method of Hennessey were applied to Webb’s invention, the result would not teach the limitations of claim 7. Instead, they would result in a change in the way the data cache, not the store queue, was organized and indexed.

In the Response to Arguments section of the Office Action mailed November 28, 2006, the Examiner disagrees with Applicants’ argument and asserts that the additional index is generated in Webb, as previously discussed and “Consequently, an obvious rejection with an analogous and extremely common technique is reasonable in this case.” Applicants again assert that Webb does not teach the additional index value of Applicants claims (i.e., one generated and used as recited in claims 5 and 6) and that there is no suggestion in the cited art that the bypass mechanism of Webb would benefit from a smaller cache size (or a smaller size for any of the data stores or queues described.) Therefore there would be no reason to apply this “extremely common technique” to the system of Webb, nor would applying it result in the limitations recited in claim 7.

For at least the reasons above, the rejection of claim 7 is not supported by the cited art and removal thereof is respectfully requested. Claim 26 includes limitations

similar to claim 7, and so the arguments presented above apply with equal force to this claim, as well.

Applicants assert that numerous other ones of the dependent claims recite further distinctions over the cited art. Applicants traverse the rejection of these claims for at least the reasons given above in regard to the claims from which they depend. However, since the rejections have been shown to be unsupported for the independent claims, a further discussion of the dependent claims is not necessary at this time. Applicants reserve the right to present additional arguments.

CONCLUSION

Applicants submit the application is in condition for allowance, and prompt notice to that effect is respectfully requested.

If any extension of time (under 37 C.F.R. § 1.136) is necessary to prevent the above-referenced application from becoming abandoned, Applicants hereby petition for such an extension. If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5500-89400/RCK.

Respectfully submitted,

/Robert C. Kowert/

Robert C. Kowert, Reg. #39,255
Attorney for Applicant(s)

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8850

Date: February 28, 2007